Office Action Summary	Application No.	Applicant(s)	
	10/594,799	YOAV ET AL.	
	Examiner	Art Unit	
	KENNETH TANG	2196	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *20 January 2011 and 15 March 2011*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,4-10,12-17 and 19-21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,4-10,12-17 and 19-21* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *3/15/11*.
4) ☒ Interview Summary (PTO-413) Paper No(s)/Mail Date. *4/5/11*.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-21 are presented for examination.

2. This action is in response to the Amendment/Remarks on Applicant's arguments filed 1/20/11 and the IDS submission on 3/15/11. Applicant's arguments have been fully considered but are moot in view of the new grounds of rejections.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

3. Claims 15-18 are directed to non-statutory subject matter.

4. As per claim 15, it is directed to a system that comprises a plurality of environments, a program, a host platform and a translator, which are software, per se. Such a claim that is directed to a software system fails to fall under one of the four statutory categories of invention under 35 USC 101. Claims 16-18 are also rejected for being dependent on rejected claim 15 and for failing to cure its deficiencies.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-2, 4-5, and 19-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kissell (US 2005/0120194 A1) in view of Kerly (US 2003/0126313 A1).

6. Kerly was cited in the IDS on 3/15/11.

7. As per claim 1, Kissell teaches a computer implemented method comprising:

beginning initialization a first thread from a second context, wherein the first thread is executable on a first context and the second context (Abstract; [0049]);

suspending the initialization of the first thread at a position within the second context, wherein the beginning initialization of the first thread is suspended in response to a detection of an operating system request to create the first thread (suspending occurs from the interrupt of the context switching, etc.) ([0007]; lines 28-32 of [0008]);

creating a second thread based on the position in the second context ([0033]; [0042]; [0043]); and

completing the initialization of the first thread continuing from the position in the second context ([0033]; [0042]; [0043]).

Kerly teaches suspending the initialization wherein the beginning of the first thread is suspended in response to a detection of an operating system request to create the first thread (see Abstract; Fig. 3). Kissell and Kerly are analogous art because they are both in the same field of endeavor of a multithreaded programming environment. Thus, it would have been obvious to

one of ordinary skill in the art at the time the invention was made to modify Kissell such that it would include the feature of suspending the initialization wherein the beginning of the first thread is suspended in response to a detection of an operating system request to create the first thread, as taught and suggested by Kerly. The suggestion/motivation for doing so would have been to provide the predicted result of reducing thrashing in a multithreaded environment ([0002]).

8. As per claim 2, Kissell teaches wherein the beginning initialization of the first thread includes allocating per-thread context resources (Fig. 2, item 226; [0030]; [0031]).

9. As per claim 4, Kissell teaches wherein the second context is the platform-independent code to be executed on a host platform (JAVA, etc.) ([0068]; [0069]).

10. As per claim 5, Kissell teaches wherein the second context is a host platform that supports multiple instruction set architectures (ISA) (MIPS32 or MIPS64 Instruction Set Architectures, etc.) ([0031]).

11. As per claim 19, Kissell teaches a computer system for managing thread resource comprising:

a random accessed memory (System memory 108) (Fig. 1);

a first processor configured to execute multithreaded programs stored in the random accessed memory (multithreaded microprocessor 102) (Fig. 1);

a multithreaded program to be executed (Abstract); and

a program to transparently initialize and create a thread included in the multithreaded program in an environment supported by the first processor, wherein the initialization of the thread is suspended in response to a detection of an operating system request to create the thread ([0007]; lines 28-32 of [0008]).

12. Kissell does not explicitly disclose a second processor. However, one of ordinary skill in the art would find it obvious that Kissell's multithreaded microprocessor is capable of containing multiple logical processors. The suggestion/motivation for executing on a plurality of logical processors would be to provide the predicted result of increasing parallelism.

13. Kerly teaches suspending the initialization wherein the beginning of the first thread is suspended in response to a detection of an operating system request to create the first thread (see Abstract; Fig. 3). Kissell and Kerly are analogous art because they are both in the same field of endeavor of a multithreaded programming environment. Thus, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Kissell such that it would include the feature of suspending the initialization wherein the beginning of the first thread is suspended in response to a detection of an operating system request to create the first thread, as taught and suggested by Kerly. The suggestion/motivation for doing so would have

been to provide the predicted result of reducing thrashing in a multithreaded environment ([0002]).

14. As per claim 20, Kissell teaches wherein the program further allocates per-thread context resources (Fig. 2, item 226; [0030]; [0031]).

15. As per claim 21, Kissell teaches wherein the program initializes and creates the thread transparently by associating the allocated per-thread context resource between the environment supported by the first processor (Fig. 2, item 226; [0030]; [0031]).

16. Claims 6-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kissell in view of Kerly, and further in view of Venstermans et al. (hereinafter Venstermans) ("64-bit versus 32-bit Virtual Machines for Java", September, 15, 2005).

17. As per claim 6, Kissell does not expressly teach wherein completing the initialization of the first thread includes executing an application programming interface that makes an operating system request to create the first thread in the second context. However, Venstermans teaches wherein completing the initialization of the first thread includes executing an application

programming interface that makes an operating system request to create the first thread in the second context (page 5, 2nd paragraph). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Kissell and Venstermans because it would provide the predicted result of a means to communicate between two different operating environments (Venstermans – Abstract).

18. As per claim 7, Kissell does not expressly teach wherein the first context is an IA-32 multithreaded program and the second context is an IA-64 multithreaded program. However, Venstermans teaches wherein the first context is an IA-32 multithreaded program and the second context is an IA-64 multithreaded program (see Abstract). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Kissell and Venstermans because it would provide the predicted result of a means to communicate between two different operating environments (Venstermans – Abstract).

19. As per claim 8, it is rejected for similar reasons as stated in the rejection of claim 7.

20. As per claim 9, it is rejected for similar reasons as stated in the rejection of claim 1. However, Bissell does not expressly teach a foreign thread is to be executed on a foreign platform. But Venstermans teaches a different thread to be executed on a different platform (Abstract). It would have been obvious to one of ordinary skill in the art at the time the

invention was made to combine the teachings of Kissell and Venstermans because it would provide the predicted result of a means to communicate between two different operating environments (Venstermans – Abstract).

21. As per claim 10, Kissell (Fig. 2, item 226; [0030]; [0031]) and Venstermans (Abstract) teaches wherein the beginning initialization of the foreign thread includes allocating per-thread context resources.

22. As per claim 11, Kissell ([0007]; [0008]) and Venstermans (Abstract) teaches wherein the beginning initialization of the foreign thread is suspended in response to a detection of an operating system request to create the foreign thread.

23. As per claim 12, Kissell ([0068]; [0069]) and Venstermans (Abstract) teaches wherein the host platform supports platform-independent code.

24. As per claim 13, Kissell ([0031]) and Venstermans (Abstract) teaches wherein the host platform supports multiple instruction set architectures (ISA).

25. As per claim 14, Venstermans teaches wherein the foreign platform is a IS-32 platform and the host platform is a IS-64 platform (Abstract).

26. As per claim 15, Kissell teaches a computer system for managing thread resource comprising:

a first multithreaded programming environment ([0031]);

a second multithreaded programming environment ([0031]);

a multithreaded program including a first thread and a second thread ([0025]; Abstract);

a host platform [0068]; [0069]; and

Kissell does not expressly teach a dynamic binary translator to manage and support the first thread for the first multithreaded programming environment to be executed in the second multithreaded programming environment. However, Venstermans teaches a computer system that utilizes a Java Virtual Machine to provide a dynamic binary translator to manage and support the first thread for the first multithreaded programming environment (32-bit mode, etc.) to be executed in the second multithreaded programming environment (64-bit mode, etc.). (see Abstract). It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Kissell such that it would include the feature of a dynamic binary translator to manage and support the first thread for the first multithreaded programming environment to be executed in the second multithreaded programming environment, as taught and suggested in Venstermans. The suggestion/motivation for doing so would have been to provide the predicted

result of taking advantage of the Java language's benefit of it platform independence, making it useful in a lot of technologies ranging from embedded devices to high-performance systems (Abstract).

27. As per claim 16, Venstermans teaches further comprises a first component to provide a communication interface between the dynamic binary translator and the multithread programming environment (page 5, 2nd paragraph).

28. As per claim 17, Kissell ([0007]; [0017]) in view of Venstermans (page 5, 2nd paragraph) teaches further comprises a first thread library and a second thread library.

29. As per claim 18, Kissell teaches further comprises a second component to intercept service requests from the multithreaded program (interrupt of the context switching, etc.) ([0007]; [0008]).

30. Claims 15-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wang et al. (hereinafter Wang) (US 2006/0184920 A1) in view of Kerly.

31. As per claim 15, Wang teaches a computer system for managing thread resource comprising:

a first multithreaded programming environment, executed by a processor ([0042]; Abstract);

a second multithreaded programming environment, executed by the processor ([0042]; Abstract);

a multithreaded program, executed by the processor, including a first thread and a second thread ([0042]);

a host platform ([0002]; [0006]; [0007]); and

a dynamic binary translator to translate the first thread for the first multithreaded programming environment to be executed in the second multithreaded programming environment, wherein the binary translator further includes an operating system wrapper configured ([0029]; Fig. 1, item 112; [0040]; [0008]).

32. Kerly teaches suspending the initialization wherein the beginning of the first thread is suspended in response to a detection of an operating system request to create the first thread (see Abstract; Fig. 3). Wang and Kerly are analogous art because they are both in the same field of endeavor of a multithreaded programming environment. Thus, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Wang such that it would include the feature of suspending the initialization wherein the beginning of the first thread is suspended in response to a detection of an operating system request to create the first

thread, as taught and suggested by Kerly. The suggestion/motivation for doing so would have been to provide the predicted result of reducing thrashing in a multithreaded environment ([0002]).

33. As per claim 16, Wang teaches further comprises a first component to provide a communication interface between the dynamic binary translator and the multithread programming environment (Fig. 1, item 109).

34. As per claim 17, Wang teaches further comprises a first thread library and a second thread library ([0042]).

35. As per claim 19, Wang teaches a computer system for managing thread resource comprising (see Abstract):

a random accessed memory (RAM 1106, Fig. 11);

a first processor capable of executing multithreaded programs stored in the random accessed memory (Processor 1102 executing multithreaded programs stored in RAM 1106, Fig. 11);

a multithreaded program to be executed on a second processor ([0042]; Fig. 11, items 1102, 1144, etc.); and

a program to transparently initialize and create a thread included in the multithreaded program in an environment supported by the first processor, to be executed on the second processor ([0042]; Fig. 11, items 1102, 1144, etc.; [0007]; lines 28-32 of [0008]).

36. Kerly teaches suspending the initialization wherein the beginning of the first thread is suspended in response to a detection of an operating system request to create the first thread (see Abstract; Fig. 3). Wang and Kerly are analogous art because they are both in the same field of endeavor of a multithreaded programming environment. Thus, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Wang such that it would include the feature of suspending the initialization wherein the beginning of the first thread is suspended in response to a detection of an operating system request to create the first thread, as taught and suggested by Kerly. The suggestion/motivation for doing so would have been to provide the predicted result of reducing thrashing in a multithreaded environment ([0002]).

Response to Arguments

37. During patent examination, the pending claims must be "given their broadest reasonable interpretation consistent with the specification." In re Hyatt, 211 F.3d 1367, 1372, 54 USPQ2d 1664, 1667 (Fed. Cir. 2000). Applicant always has the opportunity to amend the claims during prosecution, and broad interpretation by the examiner reduces the possibility that the claim, once issued, will be interpreted more broadly than is justified. In re Prater, 415 F.2d 1393, 1404-05, 162 USPQ 541, 550-51 (CCPA 1969).

38. Applicant's amendment of claim 15 fails to cure the deficiencies of 35 USC 101. Even though the amendment to the claims has the software system executed by a processor, the system is not directed to the hardware processor. Instead the system is directed to the environments, program, host platform, and translator, which are all software components. Therefore, the rejection of claims 15-18 with respects to 35 USC 101 are maintained.

39. Applicant's arguments with respect to prior art have been fully considered but moot in view of the new grounds of rejections.

Conclusion

Applicant's submission of an information disclosure statement under 37 CFR 1.97(c) with the fee set forth in 37 CFR 1.17(p) on 3/15/11 prompted the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 609.04(b). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event,

however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to KENNETH TANG whose telephone number is (571)272-3772. The examiner can normally be reached on 9:00AM - 5:30PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Emerson Puente can be reached on (571) 272-3652. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Kenneth Tang/
Primary Examiner, Art Unit 2196